

Abstract

Hamming code [1] is one of the efficient error coding algorithms widely used in wireless data communication. Existing decoders are mostly based on hardware, suffering from critical bottlenecks such as scalability, programmability and flexibility. Of late, software defined radio (SDR) [2] is a promising technology that implements a range of communication protocols in software by using central processing units (CPUs) and graphics processing units (GPUs). To satisfy the increasing demand of photorealistic graphics, real time communication service, modern smart-phones, tablets etc utilize heterogeneous multi-core SoC based processor that integrates CPU and other application specific accelerator like GPU and DSP [3, 4]. In this paper, we present a computationally efficient implementation of a Hamming code decoder for variable *error tolerance* (number of bits that can be corrected), t bits, and variable *packet size*, M Bytes, on a mobile GPU for high speed wireless communication. GPU offers an extremely high-throughput parallel computing platform by employing hundreds of processors concurrently. However, the Hamming algorithm is challenging to parallelize effectively on a GPU because it executes on sparsely located data items with several conditional statements, leading to non-coalesced, long latency global memory access and huge thread divergence. In spite of this, the proposed approach provides insights into how to produce a high-performance GPU implementation. When executed on a 336 core GPU, the achieved speedup is $99\times$ over an equivalent CPU implementation. Moreover, the implementation yields a significant reduction in computational complexity from $O(n)$ of the sequential algorithm to $O(1)$ of the GPU-based approach. Furthermore, the GPU based decoder exceedingly outperforms the CPU based approach in terms of energy efficiency.

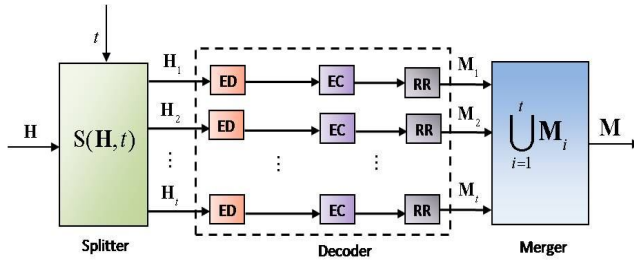


Figure 1. Components of a Hamming code decoder.

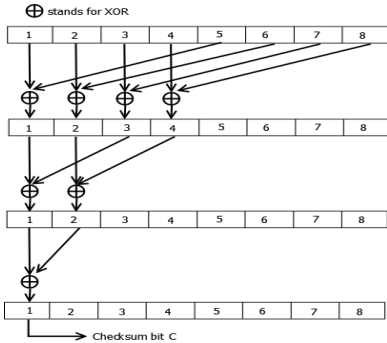


Figure 3. Checksum calculation of Fig. 2

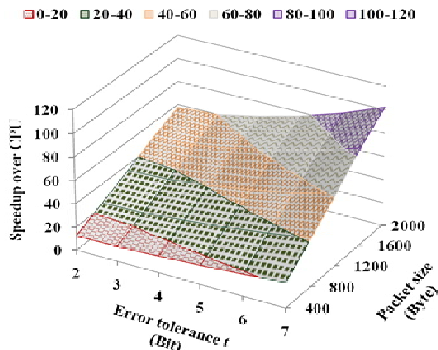


Figure 4. Speedup by GPU over CPU

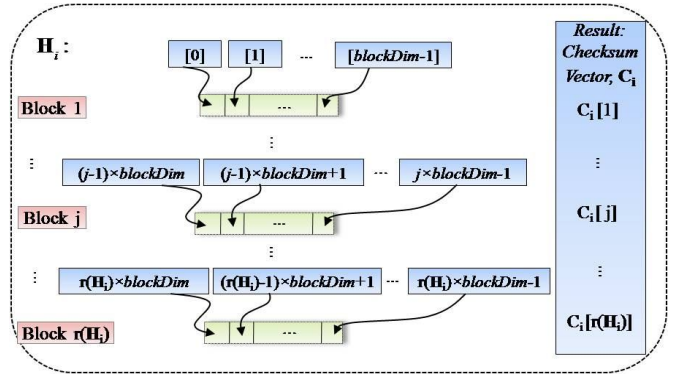


Figure 2. Decoder Implementation on GPU

$$c_j = \prod_{\forall i \in I_j} x_i, \forall j \in [0, |C|-1]$$

$$I_j = \bigcup_{i \geq 0, \forall k \in [0, 2^{(|C|-1)}]} \left\{ (2^{(j-1)}(2i+1) + k) \leq |H| \right\}, \forall j \in [0, C-1]$$

Table 1. Computational complexity

Parameter Variation	Platform	Function unit		
		Splitter	Decoder	Merger
M	CPU	$O(n)$	$O(n)$	$O(n)$
	GPU	$O(1)$	$O(1)$	$O(1)$
t	CPU	$O(\log_2 n)$	$O(\log_2 n)$	$O(\log_2 n)$
	GPU	$O(1)$	$O(1)$	$O(1)$

Table 2. Energy consumption reduction by GPU over CPU

Packet Size (Byte)	Error strength, t, (Bit)				
	t=2	t=3	t=4	t=5	t=6
M=100	87%	88%	89%	90%	92%
M=800	94%	94%	95%	96%	96%
M=1200	96%	97%	97%	97%	98%
M=1600	98%	98%	98%	98%	99%
M=2000	98%	98%	99%	99%	99%

1. R.Ma, S. Cheng, "The Universality of Generalized Hamming Code for Multiple Sources," IEEE Transactions on Communications, vol.59, no.10, pp.2641-2647, Oct. 2011.
2. P. Solic, J. Radić, and N. Rozic, "Software defined radio based implementation of RFID tag in next generation mobiles," IEEE Transactions on Consumer Electronics, vol.58, no.3, pp.1051-1055, Aug. 2012.
3. Won-Jong Lee; Youngsam Shin; Jaedon Lee; Jin-Woo Kim; Jae-Ho Nah; Hyun-Sang Park; Seokyeon Jung; Shihwa Lee, "A novel mobile GPU architecture based on ray tracing," Consumer Electronics (ICCE), 2013 IEEE International Conference on , vol., no., pp.21,22, 11-14 Jan. 2013
4. Yi-Chu Wang; Kwang-Ting Cheng, "Energy and Performance Characterization of Mobile Heterogeneous Computing," Signal Processing Systems (SiPS), 2012 IEEE Workshop on , vol., no., pp.312,317, 17-19 Oct. 2012